

Working With Sockets

socket() System Call

Prototype

```
int socket (int domain, int socket_type, int protocol);
```

Domain Macros

AF_INET

AF_INET6

AF_UNIX

AF_PACKET

...

Socket Types Macros

SOCK_STREAM

SOCK_DGRAM

SOCK_RAW

Address Conversion Functions

Prototypes

- `int inet_pton(int addr_fam1, const char *src, void *dst);`
- `const char *inet_ntop(int addr_fam1, const void *src, char *dst, socklen_t size);`

Address Families

- `AF_INET`
- `AF_INET6`

Network Byte Order Conversion

Prototypes

- `uint16_t htons(uint16_t hostshort);`
- `uint32_t htonl(uint32_t hostlong);`
- `uint16_t ntohs(uint16_t netshort);`
- `uint32_t ntohl(uint32_t netlong);`

IPv4 Address Family Struct

```
struct sockaddr_in {  
    sa_family_t    sin_family;  
  
    in_port_t      sin_port;  
  
    struct in_addr  sin_addr;  
  
    unsigned char   sin_zero[8];  
  
};  
  
struct in_addr {  
  
    uint32_t s_addr;  
  
};
```

https://docs.google.com/presentation/d/1gKfyxuK5FoAMtbuNUm_Rzqfh8aviRfm7AeuKu2Etb8s/edit?slide=id.g38024501de5_0_50#slide=id.g38024501de5_0_50

Binding the Socket

Prototype

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

Structs

```
struct sockaddr {  
    sa_family_t sa_family;  
    char sa_data[14];  
};
```

Type Definitions

```
typedef unsigned int socklen_t;
```

Listening to Connections (By First Host/Server)

Prototype

```
int listen(int sockfd, int backlog); // Backlog ----> SOMAXCONN
```

Accept the Connection (By First Host/Server)

Prototype

```
int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

Connect with Socket (By Second Host/Client)

Prototype

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

send() System Call Flags

Prototype

```
ssize_t send(int sockfd, const void *buf, size_t len, int flags);  
ssize_t sendto(int sockfd, const void *buf, size_t len, int flags,  
               const struct sockaddr *dest_addr, socklen_t addrlen);
```

Flags

- MSG_DONTWAIT // Non Block working (Same for sendto)
- MSG_MORE // Packet Is not Ended (Same for sendto)
- MSG_NOSIGNAL // Don't sent SIGPIPE (Same for sendto)
- MSG_OOB // Send Special Data (Same for sendto)
- MSG_WAITALL // Wait Until All Data Will be Sended (Same for sendto)

recv() System Call Flags

Prototype

```
ssize_t recv(int sockfd, void *buf, size_t len, int flags);  
ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags,  
                 struct sockaddr *src_addr, socklen_t *addrlen);
```

Flags

- MSG_DONTWAIT // Non - Block (Same for recvfrom)
- MSG_OOB // Receive Special Data (Same for recvfrom)
- MSG_PEEK // Receive but not clear from stream (Same for recvfrom)
- MSG_WAITALL // Full info will Receive (Same for recvfrom)
- MSG_TRUNC // Return size of packet (Same for recvfrom)